"Express Mail" mailing label no. EE785559358US

Date of Deposit: 23 May 2001
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the

Box Patent Application
Commissioner for Patents
Washington, D.C. 20231

Jeffrey Slusher
(Typed or printed name of person mailing paper or fee)

Jeffrey Slusher
Signature of person mailing paper or fee)

Atty. ref: Spotfire04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Attached is a new U.S. Utility Patent Application for:

TITLE: GRAPHICAL INPUT SYSTEM AND METHOD FOR COMPUTERS

INVENTORS:

1) Name: Martin B. NILSSON
Residence: Cambridge, Massachusetts, USA
Address: 12 Merrill Street, Cambridge, MA 02139
Citizen of: Sweden

2) Name: Anders S. FASTH
Residence: Göteborg, Sweden
Address: Landsvägsgatan 26, 413 04 Göteborg, Sweden
Citizen of: Sweden

SMALL ENTITY: YES

Graphical Input System and Method for Computers

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] This invention relates to a control mechanism in a graphical user interface, in particular, to a method for specifying values in displayed icons.

DESCRIPTION OF THE RELATED ART

[0002] The graphical user interface (GUI) has revolutionized computing. Rather than having to learn a large set of commands and key patterns, GUI users can quickly launch applications, scroll between pages, change fonts, enter data base queries, and perform many other previously tedious tasks by simply moving a mouse or similar input device in order to maneuver an on-screen cursor and "clicking" on or dragging a selector displayed on a monitor. Selectors include simple check boxes, scroll bars and buttons, icons (such as an image of a small printer to execute a "Print" command), pointers, etc.

[0003] Often, however, existing selectors can themselves be too much of a good thing: By reducing the need for the keyboard, GUI designers have also often made it more difficult, even for the most dexterous using the most sensitive mouse, trackball, touchpad, touchscreen, etc., to precisely position the selectors and enter exact values. This difficulty is of course amplified for those who are not as nimble, or who use input devices such as small touchpads on portable computers that are less precise to begin with. For example, the most commonly used word-processing applications display a graphically movable button on a scrollbar as a way for the user to select which page (or portion of a page) of a document is displayed. If the user is currently viewing, say, page 3, and wants to see page 6 instead, then he positions the cursor on the button and drags it downward until page 6 is displayed.

[0004] The problem is that if the document is large, then a slight movement of the mouse, and thus cursor, can cause a large jump in which page is displayed. The conventional solution to this is for the user to enter the desired page number in a field in

a window that he must open by clicking through one or more layers of a toolbar or pull-down menu. When using Microsoft® Word 97, for example, selecting a specific page to scroll to in this way requires the user to re-position the cursor four separate times (to the "Edit" menu, to the "Go To" menu item, to the "Enter page number" field, and to the "Go To" button icon), with three clicks and one keyboard entry of the desired page number.

[0005] The process can be shortened by using the "hot key" combination "Ctrl+G," but even this involves (for most users), releasing the mouse to reach the "G" key with the right hand while holding down the "Ctrl" key with the left, then once again reaching for the mouse to position (and click) the cursor in the "Enter page number" field, entering the desired page number, then moving the cursor to the "Go To" button and clicking once again. Even this "shortcut" therefore requires one simultaneous, double keyboard entry (Ctrl+G), one keyboard data entry (the page number), and two repositionings of the cursor.

[0006] The need to make it easier for users to adjust graphical range and value selectors arises in may other applications than word-processing; indeed, some other applications may be even more demanding. For example, there are many applications that visualize relationships between different data sets and allow users to manipulate the display of database analyses. In these applications, display devices – sometimes referred to as "graphical query devices" – translate cursor movements into search queries, for example, expressed in the common SQL language.

[0007] Thus, rather than having to enter the value "123.4" as a range limit into some portion of a command line, the user can simply maneuver his mouse and thus a screen cursor to drag a displayed range slider device until the edge of the slider aligns with the desired value, which is indicated on a scale or in a call-out displayed next to the slider. In these applications, there will typically be an even larger range of more finely subdivided possible values than there are pages in a typical document, which means that it is even more difficult – and therefore also slower – for a user to position the cursor and drag the displayed input device just right.

[0008] A similar operation is required to adjust other parameters using graphical input devices. For example, many applications, including data visualization programs and image editing applications, have a graphical device that allows the user to change a zoom level of a display.

[0009] What is needed is a way for users to be able to enter desired values in the context of a graphical user interface that requires fewer steps and less hand motion, that relies less on the precision of the mechanical input device than do existing GUI's, and that is more accurate. At the same time, users should not be required to adjust to using an unfamiliar graphical input device; rather, preferably, the full, familiar functionality of the input mechanism should still be available to the user. This invention provides such an input mechanism.

SUMMARY OF THE INVENTION

[0010] The invention provides a method and system for inputting at least one parameter into a computer. For at least one input parameter, for example, a query parameter in a database analysis routine, at least one associated primary graphical input device is displayed on a display. The system senses user selection of the primary graphical input device, which has a state that corresponds to a value of the respective input parameter and is graphically controllable by a user via at least one predetermined primary input action.

[0011] Associated with the primary graphical input device is at least one predetermined non-graphical, secondary input action that corresponds to secondary input by the user of the value of the respective input parameter. While the primary graphical input device is selected, the system senses any of the primary as well as any of the secondary input actions of the user; interprets the sensed input action of the user as input data; and sets the value of the input parameter to correspond to the input data.

[0012] In the preferred embodiment of the invention, a secondary graphical input device is generated on the display upon sensing user initiation of any secondary input action. Data entered by the user as the secondary input action is displayed within the secondary graphical input device.

[0013] Where a plurality of parameters to be adjusted, the values of the plurality of parameters are preferably associated with positions of a corresponding plurality of adjustable displayed portions of a single primary graphical input device. A respective activation region of the primary graphical input device is then preferably associated with each displayed portion. When the user selects one of the activation regions, and upon user initiation of a secondary input action, the system sets the value of the parameter associated with the selected activation region equal to data entered by the user.

[0014] A single activation region of the primary graphical input device may also be associated with a plurality of the parameters. When the system senses that the user has selected the activation region, it also senses entry by the user of a plurality of input values via secondary input action. The system then sets the values of the respective parameters according to the input values.

[0015] In one embodiment of the invention, a number of values input by the user into the secondary graphical input device is compared with the number of parameters associated with corresponding displayed portion of the primary graphical input device. If the number of values input is greater than the number of parameters, then an adjustable displayed portion of the primary graphical input device is subdividing into a number of displayed adjustable portions corresponding to the number of values input; if the number of values input is less than the number of parameters, then corresponding ones of the adjustable displayed portions are joined.

[0016] One example of a primary graphical input device is a page-selection scroll bar of a word-processing program. In this case, the parameter will in many implementations be a page number.

[0017] In several embodiments of the invention, the input data and input parameter are alphanumeric strings.

[0018] One example of how primary user input actions are performed is by maneuvering a cursor-control device such as a mouse or trackball. Secondary user input actions may then be performed using an alphanumeric input device such as a keyboard. The user may then select the primary graphical input device by maneuvering a non-alphanumeric, cursor-control device to position an on-screen cursor on the

primary graphical input device; the user then performs the primary input actions using the non-alphanumeric, cursor-control device and the secondary input actions using an alphanumeric input device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] Figure 1 is a block diagram that illustrates the main hardware and software components of a computer suitable for implementing the invention.

[0020] Figure 2 illustrates a single-entry graphical input device according to the invention as it could be used, for example, to select displayed pages of a multi-page document or file.

[0021] Figures 3A-3C show three different stages of data entry using one example – a one-sided range slider -- of a graphical input device according to the invention.

[0022] Figures 4A and 4B show two stages of data entry using another example – a two-sided range slider -- of a graphical input device according to the invention.

[0023] Figures 5A and 5B show two stages of data entry using still another example of a graphical input device according to the invention, in this case, a non-linear range indicator, namely.

[0024] Figures 6A and 6B multiple parameter adjustment using a graphical input device according to the invention.

[0025] Figures 7A and 7B illustrate the use of the invention to adjust a two-dimensional graphical input device.

DETAILED DESCRIPTION

[0026] The invention provides a way for a user to quickly and easily perform, in the context of a graphical user interface (GUI), such operations as entering values into a data-analysis or display-control application information or navigating a highly graphical display, using text commands or alphanumeric data, but still primarily through use of graphical input/display devices. Below, the main hardware and software components in a system according to the invention are described. Then, the general inventive method is outlined, after which specific examples of the many input devices according to the invention, and examples of their uses, are illustrated and discussed

MAIN SYSTEM COMPONENTS

[0027] Figure 1 illustrates the main components of a computer suitable for implementing the invention. A hardware platform 100 includes one or more processors 102, various input/output ports 104, system memory 106, a graphics card 108, and any other conventional components as needed in any given system. One or more data storage devices such as a disk 120 are also normally included.

[0028] Various physical (as opposed to purely graphical), user-operated I/O devices are connected to respective (or shared) input ports. For example, in Figure 1, a keyboard 130 and a mouse 132 are shown. The operation of such physical I/O devices is very well known – indeed, modern life in almost all workplaces and even homes is usually lived with the background sounds of fingers pressing keyboard keys and clicking mouse buttons 133, 134. Instead of a keyboard, other equivalent devices may be used for symbolic (including alphanumeric) and/or functional (such as the standard keyboard function keys F1-F12, "Ctrl" or "Alt" or "$" keys, etc.) input; these include keyboard displays on active display screens that allow a user to select "keys" by touching the screen with a finger or stylus, as well as those that allow the user to use the stylus to "write" alphanumeric information by hand on a portion of the screen. Voice-recognition software may also replace or augment the keyboard, so that the user can enter alphanumeric data into the computer without touching anything at all.

[0029] Similarly, many other devices such as trackballs, touchpads, joysticks, etc., often take the place of the illustrated mouse 132. Moreover, in some systems, such as handheld computers (including those referred to as "personal digital assistants), the user selects a current position on a display screen by touching the screen with a stylus. In still other systems, in which the display is a touchscreen, the user can activate on-screen GUI devices simply by touching the screen. Voice-recognition software also often includes the ability to control an on-screen cursor, select icons, etc., using voice commands. Any keyboard or equivalent, or any mouse or equivalent, may be used to implement the invention – the only requirement is that the user should be able to designate some current active point or region (for example, by controlling the position of a cursor 140 or other displayed position indicator using a mouse) on the display screen 142 of a monitor 144, laptop or handheld computer system, and to enter symbolic (including alphanumeric, or just alphabetical or numerical) and/or functional data. Furthermore, one can also use this invention even in systems, for example, those whose GUI is operated by a stylus and a touchscreen, that do not have a displayed cursor. The terms "keyboard," "mouse," and "cursor" are therefore used here as collective terms and only by way of example because these are the most common input devices and are intended to refer to their respective equivalent physical and graphical devices as well, several of which are mentioned above.

[0030] As is any other computer there is system software, in particular, an operating system 150, which is required to support the applications, such as application 160, that are installed to run on the computer. As in other conventional computers, the operating system 150 includes drivers 152 as necessary to act as an interface between the given system software and any peripheral devices. These peripheral devices will generally include the keyboard 130, the mouse 132, and usually the mass storage disk 120 itself, which are connected to the hardware platform. Depending on which system hardware and software are included in any given implementation of the invention, an application program interface (API) 154 is also included within the operating system. The nature, design and operation of drivers and API's are well known in the art of computer science and are therefore not discussed further here.

[0031] One form of interface that has particular relevance in the context of this invention is the graphical user interface (GUI) 156, which is typically a component of the operating system, but may also include specific hardware support as needed. GUI's are also well-understood components of modern computer systems and are therefore not explained in greater detail here. The only aspects of a GUI that are relevant to an understanding of the invention are 1) the ability, together with the drivers for the various input devices and related conversion software, to relate the motion of some physical input device (such as the mouse 132) or other user action (such as touching the display screen or entering a voice command) to the coordinate system used to generate a display on the screen 142; 2) to thereby sense what point in the display the user has thereby selected (for example, by positioning the cursor 140 by moving the mouse); and 3) to accept, interpret and associate with the currently selected position any input signals, such as a click of any mouse button or any keyboard entry, voice data, stylus touch, etc., made by the user.

[0032] A very common example of these GUI operations is when the user of a word-processing application moves the screen cursor over the button of a page scroll bar and, holding down a mouse button while moving the cursor downward, "drags" the button downward on the bar so as to cause the application to display some other page of the currently active document. Note that, in this case, not only is the document scrolled downward, but the position of the scroll bar button itself is continuously updated to correspond to the position of the cursor as long as the user keeps the cursor within the area of the display corresponding to the scroll bar and holds down the designated mouse button.

[0033] In order to implement these operations, the application 160 will typically include a software module 162 that relates the X-Y position of the cursor to whatever feature (for example, a displayed document, an icon, a "minimize" or "maximize" button, scroll bar, etc.) of the current display the tip of the cursor is over. This module 162 thus corresponds to a GUI component within the application itself. Indeed, when running most modern software packages, the user will typically see portions of a display that are controlled by different GUI modules. For example, when running the word-processing

application Microsoft Word on the Windows 2000 operating system, the user will see both a "Start" icon, which is used to communicate directly with the operating system, and various font-control icons such as "B" and "*I* ", which are associated with the Word application.

[0034] When an on-screen device is activated, the system accepts certain actions as valid inputs that are used to control the underlying routine. The user typically activates a graphical input device simply by placing the cursor over it (or, for example, by speaking an appropriate "Select" voice command). Each graphical input device, such as an icon, menu, scroll-bar, slider, etc., thus has associated with it one or more normal or *standard* actions that the user performs to input data to or otherwise use the feature the device is intended to control. Any other user action is either ignored or affects (acts as a valid inputs to) some other feature.

[0035] At least one input parameter is associated with each graphical input device. In the context of this invention, these parameters are not restricted to alphanumeric input such as numbers and letters and text strings, but may also be "functional," for example, "clicks" of a mouse button or function keys of a keyboard. Thus, an input parameter here means any input caused by user action that conforms to the predetermined input format of the underlying software routine.

[0036] In most existing GUI's, as long as the user keeps the cursor over an icon, the only user actions that the system interprets as being related to the feature associated with that icon is the clicking of a mouse button – any other user action is either ignored or activates some feature unrelated to the icon. For example, if the user places the cursor over a "Save" icon, then the only relevant user action will be the clicking of a mouse button (corresponding to issuing a "run" command), which is the only valid input "parameter" to this graphical input device. If the user moves the cursor (or otherwise deselects) out of the field of the device, which is a *non-standard* action for this graphical device, then the device is inactivated in the sense that the GUI will no longer interpret user actions as intended to be inputs to the corresponding routine.

[0037] Similarly, when the user places the cursor within a scroll bar or slider, the only actions that affect the routine controlled by the graphical device are the clicking of a

mouse button, or the holding down of a mouse button and a dragging motion of the mouse. If, using the prior art, the user were to enter a text string via the keyboard when the cursor is over the scroll bar, then the GUI will either ignore this *non-standard* action or assume it relates to some other feature such as text input to a currently displayed document.

[0038] In some other applications, if a user presses a scroll *wheel* of the mouse, then a special cursor is generated within the field of the scroll bar, which the user can place above or below the control button to direct the system to scroll up or down, respectively, with no need to drag a control button. Any other user action except pressing a mouse button, moving the mouse to move the special cursor up and down, or rolling the scroll wheel, that is, any *non-standard* action, causes the system to return to the state it was in before the wheel was first pressed.

[0039] Along with the other software modules that define the primary application itself (for example, data analysis or word-processing or calculation routines), the application 160 according to the invention includes a secondary input module 163. When this module 163 is activated, an input interpretation module 164 interprets data entry by the user according to the structure of a current primary graphical input device, and then applies the input data to a parameter adjustment module 165, which adjusts the parameter in whatever routine it is associated with. A device display module 166 then also changes the display of the primary graphical user input device (and/or removes the display of the secondary graphical input device) according to the user-entered data.

[0040] Still another module that is included in the preferred embodiment of the invention is a call-out generation module 167 that generates and causes to be displayed some secondary graphical input device such as a call-out or text box, for example, using an existing "tool-tip" feature of a common GUI. For the sake of simplicity, the secondary graphical input device is referred to here simply as the "call-out," which is to be understood as including any other graphical device that performs the same function.

[0041] The module 167 may be launched automatically and immediately, that is, as soon as the user positions the cursor or otherwise selects a primary graphical input

device designed according to the invention. In the preferred embodiment of the invention, however, the module 167 generates the call-out only after the user selects -- activates -- a primary graphical input device and takes any predetermined, valid secondary action -- not immediately generating the call-out reduces on-screen distraction to the user and the operation of invention becomes more transparent, less obtrusive, and retains the look and feel of the existing GUI as much as possible.

[0042] It would also be possible according to the invention not to generate any secondary graphical input device at all, but rather simply to evaluate user actions whenever the cursor is on any primary graphical input device that has a secondary graphical input device associated with it: If the user performs a primary action, then the system assumes the user action is an input according to the protocol and format of the primary device. If the action is a secondary action, then the system assumes that input to the routine normally controlled by the primary graphical device is according to the format of the secondary device. The user takes any other action -- an *invalid* action, then the system treats it as it normally would, that is, it either ignores it, uses it as input to some other routine, or simply deactivates the primary input device altogether. These options will become clearer by considering the examples of invention below.

[0043] All of the modules 162-167 may be designed and implemented using known programming techniques given the description below. Moreover, it is not necessary (although it will usually be easier) to include the modules 162-167 within the actual primary application 160 itself (for example, a data mining, data visualization, or word-processing program); rather, any or all of the modules 162-167 may be implemented within the operating system 150, or within a separate, dedicated application, as long as it is properly functionally linked, for example via remote procedure calls, to the primary application.

[0044] As will become clearer from the description below, the novel features provided by the invention are invoked when the user selects some primary graphical input device and then takes any predetermined secondary action associated with that primary device. On the other hand, when the invention is not invoked, the input interpretation module will perform its conventional functions. For example, if the cursor 140 is over a

portion of a document, and the user types a word via the keyboard, then the interpretation module 164 of a word-processing application will insert the corresponding input word in the appropriate place in the data file in memory comprising the document, and will cause the operating system to call the display driver to update the display (usually via the graphics card 108) accordingly. On the other hand, if the cursor is over a "Save" icon and the user presses ("clicks") the proper mouse button, then the application may instruct the operating system to copy the data corresponding to the document into its assigned locations in memory 106 or on disk 120.

[0045] As will become clearer below, the invention may be used to improve the ease of use of any type of application that uses display devices for data input. One of the many types of applications that will benefit especially from the invention, because they so often make great use of various displayed devices to select values and ranges, is data visualization. U.S. Patent 6,014,661 (Ahlberg , et al. , "System and method for automatic analysis of data bases and for user-controlled dynamic querying," issued 11 January 2000) discloses an example of such a data visualization system.

[0046] In these applications, often very complex databases are queried. The results of the queries are then analyzed and displayed in some visual format, usually graphical, such as a bar or pie chart, scatter plot, or any of a large number of other well-known formats. Modern analysis tools then allow the user to dynamically adjust the ranges of the displayed results in order to change and see different aspects of the analysis.

[0047] Assume, for example, that a marketing analyst is interested in examining sales figures for a particular product as a function of the age of the buyers. One possible visualization of such data is illustrated in Figure 1 as the bar chart 146. Assuming that the age groups of the buyers form the x-axis of the chart, then this product (which one might imagine to be the overpriced recordings of one the latest cacophonous, adolescent, popular music groups) becomes decreasingly popular as the buyers become older and, one hopes, develop better taste. If the analyst then wishes to concentrate on the results for some sub-set of the displayed results, he might, for example, use the cursor to maneuver and adjust a query device such as a conventional slider 148. Note that this example involves only two variables -- sales vs. age --

whereas many database analysis applications may have queries relating to a large number of different data sets. In "data mining" applications, moreover, the user may not even be aware that there is a correlative relationship between two data fields until the application makes him aware of it through the visualization.

GENERAL METHOD ACCORDING TO THE INVENTION

[0048] The invention assumes the following: Some application, such as data analysis or data mining software, or a word-processing routine, etc., generates at least one, and possibly many, on-screen primary graphical input devices, which may be of any type and need not be the same. Associated with each such graphical input device is at least one standard action, which is also referred to below as a "primary input action" or simply as a "primary action." As is explained above, a primary action is any action by the user, such as clicking a mouse button, that launches a routine normally associated with the device, that is, as it would according to the prior art. For example, with the cursor on a "zoom" icon, the only standard primary action is the pressing of a mouse button (to call up a pull-down menu or activate a slider), possibly followed by a movement of the mouse or other positioning device.

[0049] As in arrangements according to the prior art, the user (or the system itself, for example, as part of a data entry routine) selects -- activates -- one of the primary graphical input devices in any conventional manner, for example, by positioning a cursor over it, touching it with a stylus, by voice command, etc. This then becomes the "active" primary input device. According to the invention, as long as the user takes any primary action associated with the active device, then the system will perform as if the invention were not present. Thus, if the user drags a control button of a range slider, then its corresponding values will be adjusted accordingly, as usual.

[0050] When the system senses (in any conventional manner) that the user has selected the primary device, for example by a specified click of a mouse button, by stylus touch or voice command) it then also enables secondary user input to the primary device. Secondary input is signaled to the system by the user taking any predefined secondary input actions or, simply "secondary actions". If the user takes

any secondary action, then the system accepts these actions as secondary inputs to the routine associated with the primary device. For the sake of confirmation, and to make it easier for the user to see what he is doing, a secondary graphical input device is preferably displayed, along with the primary, whenever the user begins to perform some secondary action. If the user takes any other action besides a primary or secondary action, that is, an "invalid" action, then both the primary and secondary graphical input devices are preferably inactivated. These concepts will become clearer and more concrete by considering the examples given below. When secondary input is enabled, the user may input any symbolic or functional data using the keyboard as appropriate to the data type associated with the on-screen input device. The system then interprets the input data and acts accordingly, preferably adjusting the display of the primary input device to correspond to the input. Several specific examples will be given below in order to illustrate these general concepts.

EXAMPLES OF SECONDARY INPUT ACCORDING TO THE INVENTION

[0051] Figure 2 illustrates what a user might view on the display screen 142 while using a common word-processing application. Such a display will typically include both primarily text-based menu bars 202 and primarily icon-based toolbars 204, as well as status information fields 206 and the displayed representation of the multi-page document 208 itself.

[0052] A scroll bar, indicated generally as display field 220, has optional end delimiters 222, 224. By positioning the cursor 140 over either delimiter 222, 224 and holding down a mouse button, the user can direct the application to scroll the displayed document up or down. Scrolling can also be accomplished by placing the cursor over a displayed control button 226 and dragging it up or downward, whereby the display of the button will appear to move within a range field 228 of the scroll bar and the display of the document 208 will be adjusted accordingly. In some applications, such as Microsoft Word, a small window (not shown) may also be displayed adjacent to the range field to show which page is indicated by the current position of the button 226.

[0053] In Word, the display of the document is not actually updated until the user releases the mouse button. In other applications, such as Word Perfect, the display itself scrolls as the button 226 is moved. All of these features are very well known to any user of modern computer applications and are therefore not described in greater detail here. What is important to note, however, is that the user of this prior art arrangement controls the scroll bar and the resulting display entirely by maneuvering the mouse to control the cursor to adjust the button -- even if a window is displayed to show the user what the indicated page is, the user must still carefully and accurately drag the button 226 to avoid undershooting or overshooting the desired page.

[0054] According to the invention, however, whenever the system senses that the user has selected, that is, activated, some primary graphical input device, for example by maneuvering the cursor 140 into an activation field of the scroll bar 220, it accepts both primary and secondary user actions as valid inputs to the scroll bar routine, that is, the software module that determines which portion of a document is to be displayed based on the position of the button 226 on the scroll bar. In this case, and assuming by way of example that the user is using Microsoft Word 97, a primary user action would be any existing action used to control the scroll bar -- a mouse click above or below the button, or a mouse click (held) on the button followed by a dragging motion with the mouse. In the prior art, any other user action (for example, entering the number "5" via the keyboard) will either be ignored by the operating system or (for example, moving the cursor off the scroll bar entirely) will cause the scroll bar to be deactivated in the sense that further user actions will not affect the state of the scroll bar.

[0055] According to the invention, however, when the user selects the primary device (here, scroll bar), the system also interprets *secondary* user actions as valid inputs to the underlying executable routine. In this case, a secondary action will be any keyboard (or stylus, or voice) entry that indicates a scrolling instruction but that is not a primary action. In the example illustrated in Figure 2, two types of secondary actions could be allowed: 1) any numerical value that indicates, for example, which page number to display (scroll to), or a number of pages down or up to scroll to (indicated, for

example, by positive and negative integers, respectively.); and 2) a text string that the system should search for and then move the display to.

[0056] For example, as Figure 2 illustrates, once the cursor 140 is in the activation region of the secondary input device , the user can input the desired page number, such as "5," using the keyboard. The system then preferably, for the sake of confirmation, launches the call-out generation module 167 so that the call-out 230 is displayed with the entered data ("5") shown within it.

[0057] Note that it is not strictly necessary according to the invention ever to generate and display any secondary graphical device such as the call-out -- the system can simply accepts any valid secondary input and act accordingly. Some secondary graphical input device is preferably displayed, however, not only to allow the user to see what he is entering (which is especially useful where the secondary input is a text string, to avoid mis-typing), but also as a reminder that he has in fact just begun secondary input. Generation and display of the call-out 230 is therefore assumed in the examples below.

[0058] The system then, either automatically or after user confirmation (for example, clicking a designated mouse button one or more times, or pushing the "Enter" key on the keyboard), changes the active display to be page 5 without the user having to drag the button 226. At the same time, the system preferably updates the displayed position of the button 226 to correspond to the "Page 5" position it would have had if the user had dragged it there.

[0059] The system may then remove the call-out 230 from the display either immediately, or only after the user moves the cursor out of the activation region or otherwise de-selects or inactivates the primary graphical input device. In the preferred embodiment of the invention, the secondary input device 230 is removed from the screen as soon as the user performs any action other than one of the predetermined secondary actions. In order to prevent the system from displaying the secondary input device 230 unnecessarily, it is preferable to require the user to move the cursor out of any activation region and then back into it in order for a new call-out to be displayed and new data entry enabled.

[0060] In the preferred embodiment of the invention, the system accepts input -- primary and secondary -- to a primary graphical input device as long as the device is active. Thus, primary and secondary input will be enabled simultaneously. This is not necessary, however; rather, depending on the needs of a given application, the user could also be required to do something to specifically enable secondary input. For example, a user might be required to position the cursor on one or more pre-defined, dedicated activation fields associated with the primary device. This feature will be particularly useful where the primary graphical input device is designed to accept more than one input value, for example, the limits of a parameter range. In this case, the activation region(s) of the graphical input device 220 should preferably be chosen to allow the greatest ease of use of the invention by the user; this may of course vary with the application and can be determined using normal GUI design considerations. One example of an activation region for the device 220 could be the region of (that is the portion of the display lying within the coordinates for) the displayed button 226 itself, and/or anywhere in the range field 228 on either side of the button, and/or within either delimiter 222, 224.

[0061] It would also be possible to include in the display a special activation field, for example, with a dedicated icon 232 that is separate from the graphical input device 220 but functionally associated with it by the GUI. In order to make it easier to use, such a separate activation region should preferably be displayed at least near the input device with which it is associated, or near some other field or icon 234 that is related to the parameter to be input (here, page number).

[0062] If specific user action is to be required to enable user input, then arranging the corresponding activation field as a sub-region of the graphical input device itself is preferred, especially where the primary device sets more than one parameter. First, it is more intuitive to use, since the user assumes that operations relating to a graphical input device can be done by pointing at the device. Second, it does not require any persistent change to what the application displays (for example, the call-out 230 preferably disappears after the input operation) change. Third, in applications such as data analysis, there are often many primary graphical input devices, and all secondary

graphical input devices will be immediately associated with their respective primary input devices without cluttering the display screen and causing confusion. Even in common word-processing programs, for example, there will usually be at least two scroll bars active at any given time: A vertical bar for up-down page scrolling and a horizontal bar for left-right page scrolling. Fourth, the user will not need to decide before moving the cursor whether he will use primary or secondary input.

[0063] More than one activation region may be provided for a single primary device. Continuing with the scroll bar example, in Figure 2, the regions of the scroll bar above and below the button 226 could be "up" and "down" activation regions, respectively. Note that they already are interpreted differently by most existing word-processing programs: clicking in the "up" region is interpreted as a "page-up" command and clicking in the "down" region is interpreted as a "page-down" command. Such clicking is a primary action for the device. As just one example of how the invention could be configured in this application, the different activation regions could retain their directional function, but a secondary action could be enabled for each: The interpretation module 164 could then interpret entry of a text string such as "Spotfire" or "Detailed Description" above/below the button as a command to search upward/downward for the string and to change the display to show the page on which the string is found. Similarly, entering a numerical value such as "3" above/below the button could mean to scroll upward/downward by three pages.

[0064] Note that this string searching capability of the invention is not limited to use with word-processing programs. For example, when viewing a page of Internet content using a standard browser such as Internet Explorer, the only primary actions normally allowed when the cursor is over a displayed page are to operate a scroll wheel or to move the cursor to some other device field such as a "Back" icon. By designating any part of a browser display (including anywhere in the displayed page itself) as an activation field, an alphanumeric keyboard entry could be interpreted as a valid secondary action. By typing, for example, "Spotfire," the browser could then be instructed to find the next occurrence of this text string in the document, or to highlight all occurrences, etc. As long as the activation field(s) and valid secondary actions are

made known to the user, any data could be input using the invention that would otherwise require using the existing primary graphical input devices.

[0065] Figures 3A-3C illustrate the invention as it might be used in a data analysis or visualization application. In these applications, the, primary graphical input device 320 is a graphically adjustable device commonly referred to as a "slider", "alphaslider," "Range slider,", or "Item slider" which usually has the same general appearance and function as a scroll bar (which itself can be considered as a type of slider), but is usually (but not necessarily) used to select a range of values instead of just a single value. Sliders may have many different structures depending on the type of parameter (for example search query range) they are intended to control. The invention can be used with all such devices.

[0066] Of course, a full display of the results of a data analysis would normally include some form of plot or table showing the results – the slider would be used to adjust, for example, some aspect of the display, or to submit a data base query. Only the slider is shown in Figures 3A-3C merely for the sake of simplicity: It is well know in the arts of data base management, data mining, and data visualization how sliders are used to control displays and the invention may be used to advantage along with any such conventional techniques.

[0067] One difference between a typical slider 320 and a typical scroll bar is that the slider will usually have either some adjacent scale 330 or other feature to show the user what range is currently indicated. Merely by way of simple example, the scale 330 in Figures 3A-3C is a scale from 0 to 100, for example indicating a percentage; the actual scale used will of course depend on many factors, such as what data type the slider corresponds to, the distribution and number of "hits" in a data base query, etc.

[0068] The slider may also be provided with some indicator 340 that makes it easier for the user to position the cursor 140 for dragging one edge of the button. In this example, the user would normally, that is, according to the prior art, adjust the range of the slider from 0 to some value by dragging the indicator 340 to the right until the right edge of the button aligns with the desired value. In Figures 3A and 3B, the current range indicated by the slider is 0-40. In data visualization applications, for example, this

will then lead to a corresponding change in what values are displayed, what new query is submitted, etc.

[0069] Using the invention, however, the user positions the cursor 140 within a predetermined activation field of the slider, for example, by pointing to the button 326, or adjacent to some other suitable and easy-to-see feature. As Figures 3A and 3B illustrate, the activation region 328 need not be on any particular displayed feature such as the button 326, but rather may comprise any predetermined portion of the displayed device, as long as this portion is made known to the user.

[0070] The user may then enter any desired value, for example "60," using the keyboard, which the system then preferably displays within the call-out for the sake of confirmation (see Figure 3B). Figure 3A illustrates how the system may generate a blank call-out (empty input field) as soon as the user points to the activation field. As is mentioned above, however, the call-out is preferably not generated until the user begins some valid secondary action, such as entering the "6" of "60". Thus, the preferred initial configuration in this example is Figure 3B.

[0071] Once the user takes whatever predetermined confirmation action is required, if any, for example, pressing the "Enter" key or a designated mouse button, then the system accepts and interprets the entry, and adjusts the corresponding parameter and the size of the button accordingly, just as if the user had adjusted the device completely. The final state of the slider in this example is shown in Figure 3C.

[0072] The invention is not restricted to use with primary graphical input devices that provide for entry of only a single parameter. Figures 4A and 4B illustrate one of many examples of a slider 420 that allows a user to graphically adjust both the upper and lower limits of a range of values. In conventional sliders, the only valid user actions are clicking a mouse to the left or right of the button or dragging either edge. In the illustrated example, the right and left edges of the button 426 can be dragged individually to specify the upper and lower limits of an alphabetical input parameter range. As shown in Figure 4A, the current input range is J-S. According to the invention, however, these actions are only the primary actions that are associated with the slider.

[0073] A different activation region is then predetermined and associated with each parameter. For example, as in Figure 4A, the region 428L of the slider to the left of the button 436 is an activation region for the lower limit parameter and the region 428U of the slider to the right of the button 436 is an activation region for the upper limit parameter. When the user positions the cursor in either activation region, then the system enables secondary input for the respective parameter. Upon sensing such secondary action, if the system preferably generates the call-out 230 adjacent to that region, for example, pointing to the edge of the button that the user would otherwise have to drag. The user can then enter the desired parameter as before. Figures 4A and 4B show how the user adjusts the lower bound from J to E. As before, once the entry, here "E", is confirmed, the system adjusts the input range and size of the button accordingly, and preferably also stops displaying the call-out. In the example shown in Figures 4A and 4B, the input is assumed to be alphabetical. The valid secondary actions for this device would thus be an alphabetical entry using the keyboard (or stylus, or voice, etc.).

[0074] In general, to reduce the need for the keyboard, there is preferably one activation region (sub-region) for each adjustable parameter of a multi-parameter primary graphical input device. By slightly complicating the entry format, however, it would also be possible to use a single activation region and call-out to enter either or both (or all) parameters. One way to accomplish this would be to allow the user to enter multiple values separated by some predetermined separator such as a dash, semi-colon, colon, etc. Single entries could be assigned to the corresponding parameter as before, that is, as a function of the current location (activation region) of the cursor. Multiple entries, however, would be interpreted as a requested change of each parameter entered. For example, if the user enters "E-W" into the call-out 230, then the system adjusts both the lower and upper bounds of the parameter accordingly.

[0075] One other way to enable multiple entries would be for the call-out to be displayed initially with the current range(s) or value(s) of the input device. The user may then select and change any or all of the displayed parameters at once. After confirming the entry, the system can then adjust all parameters (or just all changed

parameters) accordingly. For example, in Figure 4A, the call-out could initially contain the display "J-S", using any separator such as a dash, semi-colon, colon, etc., or even separate fields – one per parameter -- which are selected using the mouse. The user could then change the call-out contents to "E-S" to cause the range change shown in Figure 4B. One advantage of this multi-entry call-out is that it allows the user to adjust more than one parameter without having to reposition the cursor into different activation sub-regions in order to change more than one parameter. This feature may be particularly valuable in applications that analyze or at least query large databases, which might otherwise execute a completely new query as soon as one parameter is changed, even though the user is interested in new results only for a change in two or more parameters.

[0076] Multiple-parameter entry (for example, into a displayed call-out) according to the invention may also be used not only to adjust the input range(s) of a given (such as the button 426), but also to change the nature and number of the adjustment features themselves. Figures 5A and 5B illustrate this possibility. Given the same slider 420 as in Figure 4A, assume the user enters multiple ranges, not just multiple values, into the call-out 230. In Figure 5A, for example the user has entered "E-J;S-W". The system then interprets this to indicate a query (or display control) having two ranges. It then also subdivides the single button 426 accordingly, as is illustrated in Figure 5B. Activation sub-regions can then be established for each of the now four separate input parameters.

[0077] Conversely, if the user enters a single range, perhaps with a null value for any other multiple ranges, the user could indicate to the system that it should join currently separate ranges. For example, given the slider as shown in Figure 5B, if the user moves the cursor out of any activation range (to reset the call-out display feature), and then back in, and then enters, for example, "J-S;" (using a semi-colon as separator but no value for the right button) the display would return to its state as shown in Figure 5A and the two ranges E-J and S-W would be replaced with the new, single range; the two buttons in Figure 5A will also be joined into a single button, corresponding to the now single value range.

[0078] Using conventional graphical input devices, these range sub-division and joining operations cannot usually be accomplished at all, or else they require several mouse clicks, dragging operations, and/or keyboard entries. Using this embodiment of the invention, the desired subdivision (or joining) of the query range(s) can be accomplished with very few keyboard entries, and actually less (and less precise) cursor motion than is required in the prior art simply to adjust a single value using a slider.

[0079] As Figures 6A and 6B show, the invention may also be used with non-linear graphical input devices. In this example, an angular range from 0 to 360 degrees is normally specified by dragging azimuth lines, which, in Figure 6B, are shown as setting the angular range from 30 degrees to 105 degrees. Displays such as these are found, for example, on radar screens, which use the azimuth lines, for example, to define alarm sectors. (Objects located within the sector cause an alarm to sound).

[0080] Using the invention, an activation region may be designated for each azimuth line, for example, anywhere near the desired line. The line itself, or some indicator (such as the tag box showing the current angle 105) could then be caused to blink, or be displayed in a different color, etc., to indicate that the cursor is within the activation zone for that line. (Similar graphical techniques could of course be used in the examples of other input devices above.) As before, the system then generates a secondary graphical input device (here, again, a call-out) adjacent to the selected azimuth line. The user can then enter a new value; the azimuth line (and the corresponding parameter) is then adjusted accordingly. In Figures 6A and 6B, for example, the user has changed the azimuth range from 30-105 to 30-60 degrees.

[0081] Figures 7A and 7B illustrate how the techniques according to the invention can be used to adjust a multi-dimensional primary graphical input device, which, in this example, is a window 726 around a sub-set of data points (shown as "x's") in an X-Y plot. In this case, there are four parameters that can be adjusted, namely, the lower and upper bounds for both the X and Y parameters. Each edge of the window 726 thus functions as a single-parameter primary graphical input device. For each, an activation region is defined in any predetermined manner and whenever the cursor is moved into

that region, secondary input is enabled (and a call-out is preferably displayed) so that the user can directly enter a new value for the corresponding parameter bound, without having to drag an edge of the window with the mouse. In Figures 7A and 7B, the activation region for the upper X limit is, for example, to the right of the window, but between the current upper and lower limits for the Y parameter. The user has changed the upper X limit from X2 to X3.

[0082] As is mentioned above, an input parameter in the sense of this invention means any input caused by user action that conforms to the predetermined input format of the underlying software routine. The input parameter(s) that may be entered using the invention are not limited to those directly associated with a particular graphical input device. Rather the invention may be used to input parameters according to predetermined heuristic rules.

[0083] As just one of many possibilities, assume that the range slider 420 (Figure 4A) is used primarily to adjust some parameter P, which is represented as a column in a data structure used in a data mining and visualization routine that generates a histogram such as is shown in Figure 1 as plot 146. As before, the user could position the cursor 140 over the slider and enter a value such as the illustrated value "E". According to this "heuristic" embodiment of the invention, the user could instead enter some other input such "4 bins" to indicate to the underlying routine that the single column used to represent P should instead be subdivided into four columns.

[0084] As another example, assume the user places the cursor over the range slider 420 and types "blue". The underlying routine could then interpret this entry not as an adjustment of the parameter P, but rather as an instruction to use the color blue to display its visualization of whatever parameter the slider 420 controls.

[0085] As still another example, assume the user of a web browser positions the cursor over a standard "Back" icon. In conventional browsers, the only acceptable primary input is the click of a mouse button (or equivalent). Using this heuristic embodiment of the invention, however, the user might instead enter via the keyboard a text entry such as "Spotfire". The browser could then interpret this as an instruction to search backward through the user's browsing history to find the first web site that

contains the word "Spotfire", for example, www.spotfire.com. Entering a number n could instead be interpreted as an instruction to go to the n'th earlier web page viewed.

[0086] As yet another example, assume that the user is working with a statistical analysis program and that this program has an on-screen toolbar icon used to select various statistical algorithms by means of some primary input parameter. Using the invention, the user could, for example, position the cursor over the icon and enter some secondary input parameter such as "4 K-mean", which the underlying program could interpret as an instruction to select for execution from among different available algorithms execute an analysis algorithm that performs "K-means clustering" with an argument of "4". Note that this is also of an example of a secondary input entry that has more than one "part," that is, that allows for entry of more than one heuristic parameter. If desired or necessary to make it easier or possible for the underlying application to parse distinguish which type of the user is attempting, prefixes such as "#" or "$" or other type (or characteristic) identifiers may also be used to indicate particular input types.

[0087] Indeed, it is also possible according to the invention to use a single primary input device for entry of different types of secondary input parameters, or even third- or fourth-order, etc. parameters. For example, combining two examples above, assume that the statistical analysis program is also the program used to generate the histogram. Assume further that the range slider's primary input parameter is a range of numerical values set by dragging the edges of the button 426. The primary input action is therefore dragging of the button, or its edges, using a mouse (or equivalent). Assume the scale of the range slider is from 0 to 100 (as in Figures 3A-3C) and that the user wishes to adjust the range of the corresponding parameter to [50,80].

[0088] The standard action to accomplish this would be to drag the left edge of the button to be aligned with the "50" scale entry and the right edge to "80". As is mentioned above, according to the invention, the user could instead place the cursor on the slider and type "50;80". This is an example of a secondary input that is of the same type (here, numeric), that is, has the same characteristic, and is used to input the same parameter(s) as the primary input device itself. Assuming the underlying GUI is so

designed and that the analysis program has such parameters, however, then according to this embodiment of the invention, the user might instead type "blue," or "4 K-mean," which the GUI then parses (using any conventional methods) and passes to the analysis routine as instructions to change the corresponding display portion to blue, or the analysis to 4-K mean clustering.

[0089] Expressed more generally, assume an underlying routine that generates and interprets a primary user action as inputting one or more value(s) Vi for primary input parameters when a graphical input device is selected. According to the invention, at least one secondary user action (which generates secondary input parameter(s)) is interpreted either as inputting the value(s) Vi, or as inputting value(s) corresponding to some other aspect or feature of the underlying routine, which may, but need not, be of the same data type or have the same distinguishing characteristic(s) as the value(s). All that is required to implement the invention in these cases is any conventionally designed parsing or type-identification routine to distinguish whether the entered value is a primary or secondary input parameter, and that the user be made aware of the predetermined semantics for and meanings of allowable secondary user actions (secondary input parameters).

[0090] In all of these "heuristic" embodiments of the invention, the graphical input device is associated with some underlying executing application. This application has one or more user-controllable, that is, user-selectable, features, such as not only the range of input parameters, but also the type of analysis to be performed, the method of display or other presentation to be used, etc. The application then, using conventional GUI technology, causes at least one graphical input device to be displayed for the user. One or more primary input parameters having first predetermined data types or structures are associated with this graphical input device and are used as inputs to set or adjust a first feature of the application.

[0091] According to the invention, however, the user is also allowed to enter at least one secondary input value by means of secondary user actions. This (these) secondary value(s) may be of the same first predetermined data type or structure, in which case they are used to control or adjust the same first feature of the application.

On the other hand, in this heuristic embodiment of the invention, the secondary input parameters may also be of a second predetermined data type or structure. When the application senses this, then it applies the input values to control or adjust some other feature of the application. For example, entering a numerical value might be interpreted as the value for a range slider used to set the value of an input parameter to a current analysis algorithm (a first feature), whereas entering "4 K-mean" might be interpreted as selecting which algorithm (a different, second feature) is to be applied.

[0092] As long as the underlying GUI for a given program is designed to determine when a position indicator such as a cursor lies within some delimited region of a display screen, the invention may also be used to "convert" an on-screen object not normally used for input into one that is. In these cases, the primary input parameter is a "null" value, or it may be one or more mouse clicks and/or mouse movements used to activate the object to allow for resizing, repositioning by dragging, or a selection from a pull-down menu of such features as deletion, borders, etc. For example, assume that the user is working with a drawing program such as a CAD application, or even with a mixed text/graphics application such as Microsoft PowerPoint. Assume further that the user positions the cursor over, for example, a display of a red circle. By typing "blue" or "new" for example, without preceding it with a mouse click, the underlying program could change the color of the circle, or create a copy of it. In cases such as these, the secondary input actions are used to control the on-screen display of the "input" device itself.